



# Plagiarism Checker X Originality Report

**Similarity Found: 29%**

Date: Friday, November 20, 2020

Statistics: 853 words Plagiarized / 2920 Total words

Remarks: Medium Plagiarism Detected - Your Document needs Selective Improvement.

---

IJARCCCE ISSN (Online) 2278-1021 ISSN (Print) 2319-5940 International Journal of Advanced Research in Computer and Communication Engineering Vol. 9, Issue 4, April 2020 Copyright to IJARCCCE DOI 10.17148/IJARCCCE.2020.9408 41 Analysis of Preemptive Shortest Job First (SJF) Algorithm in CPU Scheduling Tri Dharma Putra University of Bhayangkara Jakarta Raya Abstract: Scheduling is a key concept in computer multitasking and multiprocessing operating system design and in real-time operating system design by switching the CPU among process.

Shortest Job First (SJF) is a wellknown algorithm in CPU processing. There are preemptive Shortest Job First and non preemptive Shortest Job First Algorithms. Preemptive SJF is when the process can be interupted when a new process gets in. When with non preemptive algorithm, the algorithm will continue to be finish eventhough a new process gets in.

In this journal, we will discuss preemptive Shortest Job First Algoritm in CPU processing. An efficient Shortest Job First is with lower average waiting time and turn around time. Three case studies are discussed to understand this algorithm more deeply. Keywords: Shortest Job Frst (SFJ) algorithm, preemptive algorithm, non preemptive algorithm, turn around time, average waiting time I.INTRODUCTION Operating System (OS) is system software which acts as an interface between a user and the computer hardware.

OS is also known as resource manager because its prime responsibility is to manage the resources of computer system. Scheduling is a fundamental and most important design. Scheduling refers to set of rules, policies and mechanism that govern the order in which resources is allocated to various processes and the work is to be done.

The scheduling is a methodology of managing multiple queues of processes in order to minimize delay and to optimize performance of the system. A scheduler is an OS module that implements the scheduling policies. Its primary objective is to optimize the systems performance according to criteria set by the system designer (Shweta Jain, 2016).

The Central Processing Unit (CPU), is an important component of computer system; hence it must be utilized efficiently. This can be achieved through what is called CPU scheduling. The CPU scheduling can be defined as the art of determining which process runs on the CPU when there are multiple runnable processes.

Also, it is the problem of deciding which computer process in the ready queue (in other words, which particular programs need some processing and are ready and waiting for it) is to be allocated to the CPU for processing. It is a fundamental problem in operating system (OS). In terms of minimizing wait time for user when she or she simply wants to execute a particular set of tasks.

It is important because it has a big effect on resources utilization and the overall performance of the system (Hasan, 2014) The idea of multiprogramming is arranging multiple processes to be run by the CPU. That is when process is allocated to resources of the CPU. When more than one runnable processes exists, only one process will be allocated the resources. Algorithms for CPU resources allocation are many.

Round Robin Scheduling, FCFS (First Come First Serve) Scheduling, SRT Scheduling, Priority Scheduling, Multilevel Queue Scheduling. One of them is Shortest Job First (SJF) Scheduling Algorithm. CPU scheduling algorithms deal with the problem of deciding which process is ready queue should be allocated to CPU.

The aim of this research is to give analysis for Preemptive Shortest Job First (SJF) algorithm. Explanation and analysis is described in Gantt Chart and tables. In this algorithm every process that exists in queue will be executed based on the smallest burst time. This will result in shorter waiting time for every process, and thus the average waiting time will also becomes shorter (Monica Santika, 2014). II.CPU SCHEDULING ALGORITHM a) Scheduling Algorithms A CPU scheduler is tasked with choosing which process to run first from the ready queue.

A scheduling algorithm is used to choose the next process. Scheduling is a fundamental function of an operating system. A scheduling refers to picking a process, from many ready made processes, that is to be next executed on CPU. The CPU scheduling can be IJARCCCE ISSN (Online) 2278-1021 ISSN (Print) 2319-5940 International Journal of

Advanced Research in Computer and Communication Engineering Vol. 9, Issue 4, April 2020 Copyright to IJARCCCE DOI 10.17148/IJARCCCE.2020.9408 42 defined as the art of determining which process runs on the CPU when there are multiple runnable processes (Abbas, 2015).

FCFS (First Come First Serve) Scheduling is also termed as First In First Out, that is, allocates the CPU in order in which processes arrive to the ready queue and ready queue is maintained in a FIFO queue. The entry of a new process takes place through the tail of the queue and exit takes place through the head of queue (Kuldeep Singh Kaswan, 2017).

Round Robin Scheduling is process scheduling that implements preemptive strategy, but not preempted by other process, but mainly by dispatcher based on time allocated for processes which is called quantum. The rule is if the quantum expires and the process is not finished, then process will be switched to other process and if quantum is not finished, then process will be stopped and process is switched to other process. The only interesting issue with round robin scheduling is the length of the quantum.

Setting the quantum too short causes too many context switches and lowers the CPU efficiency. On the other hand, setting the quantum too long may cause poor response time and approximate FCFS. In any event, the average waiting time under round robin scheduling is often quite long.

For Priority Scheduling, the basic idea is straightforward, each process is assigned a priority. Equal-Priority processes are scheduled in FCFS order. SJF can be said to be Priority Scheduling, which is the longer the CPU burst, the lower the priority and vice versa. Priority can be defined internally or externally. Internally defined priorities use some measurable quantities or qualities to compute the priority of a process.

Multilevel Queue Scheduling Algorithm partitions the ready queue in several separate queues, in a multilevel queue scheduling process area permanently assigned to one queue. The processes are permanently assigned to one another, based on some property of the process, such as memory size, process priority, and process type. Algorithm chooses the process from the occupied queue that has the highest priority, and runs that process. It can be preemptive or non-preemptive.

Each queue has its own scheduling algorithm. SJF Scheduling Algorithm is especially appropriate for batch jobs for which the run times are known in advance. In this algorithm every process that exists in ready queue can be executed based on the shortest burst time. This will result in less waiting time for every process and thus

average waiting time becomes shorter, so to speak this algorithm can be said optimum.

b) Performance of Scheduling The performance of scheduling is linked to several parameters: 1) CPU Usage: CPU should be kept busy at 100% of time.. 2) Throughput: Number of processes that typically end executing in the given moment of time 3) Turnaround time: Time which is necessary for the execution of a process 4) Waiting time: Is the time that a process must wait in queue ready to be executed 5) Response time: Is the time between the reception of the request made, to the first response (Nevila Xoxa, 2014).

The following conditions should be achieved in order to make optimal scheduling: CPU usage should be maximum, throughput should be maximum, turnaround time should be minimum, waiting time should be minimum, and response time should be minimum. III. ORGANIZATION STRUCTURE This journal organizes in the following steps: We divide it into six chapters. First chapter is Introduction.

In this chapter we discuss some theories of scheduling. Second chapter is about CPU Scheduling Algorithms. Chapter three is about Organization Structure, Chapter four is about Preemptive Shortest Job First Algorithm. In chapter five we discuss about case studies, actually three case studies. The last chapter, chapter six is Conclusion.

IV. PREEMPTIVE SHORTEST JOB FIRST ALGORITHM The " Preemptive Shortest Job First " is a wellknown scheduling algorithm.

In this scheduling algorithm the ready queue is organized according to the burst time of process. The routines which requires small amount of time to its completions are placed in front of queue. This algorithm is also a preemptive scheduling algorithm. The process which is running is preempted and the new process with small burst time starts its execution. International Journal of Advanced Research in Computer and Communication Engineering Vol. 9, Issue 4, April 2020 Copyright to IJARCCCE DOI 10.17148/IJARCCCE.2020.9408 43 comes in queue then the process which is running is preempted and the new process with small burst time starts its execution.

When process successfully completes its execution then the process is terminated and removes from waiting process list. New process is selected for execution from front of ready queue (Muhammad Akhtar, 2015). The Preemptive Shortest Job First (SFJ) scheduling is especially appropriate for batch job for which the run times are known in advance.

The Shortest Job first (SFJ) algorithm gives the optimal average turn around time for a set of process, but it suffers from starvation for long processes (Mario Jean Rene, 2014).

But Preemptive SJF is not considered to be practical, because the execution time of tasks is not known in advance (Ji Young Heo, 2017) . V.CASE STUDIES In this chapter we will discuss three case studies about **Preemptive Shortest Job First** Algorithm.

Each with five processes and arrival time and also burst time. We discuss each one of it as follows: a) Case Study 1 We select five processes with their **arrival time and burst** time. These processes are, off course, are sceduled by **Preemptive Shortest Job First** algorithm. As per the table below: Table 1.

A set of five processes with time of arrival and burst time Processes Time of Arrival Burst Duration P1 0 4 P2 2 3 P3 4 6 P4 7 8 P5\_ 8 9 First job arrives at arrival time 0 is P1, at arrival time 2, system preemptive, sistem compare two process P1 and P2, P1 still with 2 and P2 still 3, so the shortest is P1, so at arrival time 2, P1 still continue.

At arrival time 4, there exists two processess, P2 and P3, the shortest is P2, that is 3 compared to P3 what is 6, so the system continue with P2 until arrival time 7. P2 finish in 7. At arrival time, 7, P4 comes up, it will be compared with P3, since P3 is shorter, 6, compares to 8, then the system chosed P4. At 8 P5 comes up, but P3 is still the shortest which is 5 second compare to P4 which is 8, and P5 which is 9. P3 continue until 13.

At 13 P4 and P5 is compared, the system will chose P4 **since it is the** shortest until 21. The last one is P5 which is 9, until 30. The processes are described in Gantt Chart as follows : P1 P2 P3 P4 P5 0 4 7 13 21 30 Gambar 1. Gantt Chart Here is the table for analysis and performance: Table 2.

Time processes and performance analysis Process Time of arrival Burst Duration Start time Finish time Trunaround time P1 0 4 0 4 4 P2 2 3 4 7 5 P3 4 6 7 13 9 P4 7 8 13 21 14 P5 8 9 21 30 22 **IJARCCCE ISSN (Online) 2278-1021 ISSN (Print) 2319-5940 International Journal of Advanced Research in Computer and Communication Engineering Vol. 9, Issue 4, April 2020 Copyright to IJARCCCE DOI 10.17148/IJARCCCE.2020.9408 44 The average waiting time is  $(0+2+3+6+13)/5 = 4.8$**

The average turnaround time is  $(4+5+9+14+22)/5 = 10.8$  b) Case Study 2 Again, we select five processes with their **arrival time and burst** time. These processes still sceduled based on **Preemptive Shortest Job First** Algorithm, as follows: Table 3. A set of five processes with time of arrival and burst time Processes Time of Arrival Burst Duration P1 0 4 P2 2 6 P3 4 5 P4 7 8 P5\_ 8 9 P1 arrives at 0, so this is the first process.

At 2, P1 will be compared with P2, since P2 arrives at 2, the remaining time of P1 is 2, compared with 6 of P2, then P1 still continue until finish, at 4. Then at 4, P3 arrives, it will

be compared with P2, since P2 is 6 and P3 is 5, so P3 will be process first until 7. At 7, P4 arrives, which will be compared with P2, and P3. P3 is left 2, so P3 continue until 8. At 8, P5 arrives, it will be compared with P3, P2 and P4.

The shortest is still P3 which is 1, so system continues until P3 finish at 9. Then system choses P2 since P2 is only 6 until finishes at 15. Then P4 starts at 15, because P4 is shorter then P5 which is 9. At 23, P5 finishes until 32. The processes are described in Gantt Chart as folows : P1 P3 P2 P4 P5 0 4 9 15 23 32 Gambar 2.

Gantt Chart Here are the table for analysis and performance: Table 4. Time processes and performance analysis

Process	Time of arrival	Burst Duration	Start time	Finish time	Trunaround time
P1	0	4	0	4	4
P2	2	6	9	15	13
P3	4	5	4	9	5
P4	7	8	15	23	16
P5	8	9	23	32	24

The average waiting time is  $(0+7+0+8+15)/5=6$  The average turnaround time is  $(4+13+5+16+24)/5= 12.4$

c) Case Study 3 Still, we discuss the Preemptive Shortest Job First Algorithm. We have five processes P1 until P5. As table belows: Table 5. A set of five processes with time of arrival and burst time

Processes	Time of Arrival	Burst Duration
P1	0	4
P2	2	3
P3	4	3
P4	7	1
P5	8	2

IJARCCCE ISSN (Online) 2278-1021 ISSN (Print) 2319-5940 International Journal of Advanced Research in Computer and Communication Engineering Vol.

9, Issue 4, April 2020 Copyright to IJARCCCE DOI 10.17148/IJARCCCE.2020.9408 45 P1 arrives at 0, so it is processed first. At 2, P2 arrives, P1 are compared with P2, the remaining time of P1 is 2, which is still lower than 3 of P2, so P1 continues until finish at 4. At 4, P3 also arrives, which is compared with P2, since the burst time of P2 and P3 are the same, sistem uses first come first serve basis, which is P2.

P2 finishes at 7. P4 arrives at 7 also, it will be compared with P3. Since P3 is 3 and P4 is 1, system will chose P4. P4 finishes until 8. At, P5 arrives, which will be compared with P3, since P5 is shorter which is 2 compared with P3 which is 3, P5 is processed until 10. At last since no other process P3 is processed until 13. The processes are described in Gantt Chart as folows : \_P1 P2 P4 P5 P3 0 4 7 8 10 13 Gambar 3.

Gantt Chart Here are the table for analysis and performance: Table 6. Time processes and performance analysis

Process	Time of arrival	Burst Duration	Start time	Finish time	Trunaround time
P1	0	4	0	4	4
P2	2	3	4	7	5
P3	4	3	10	13	9
P4	7	1	7	8	1
P5	8	2	8	10	2

The average waiting time is  $(0+2+6+0+0)/5 = 1.6$  The average turnaround time is  $(4+5+9+1+2)/5 = 4.2$  VI.CONCLUSION Preemptive Shortest Job First (SJF) is a wellknown algorithm for CPU processes scheduling. This algorithm is a scheduling algorithm.

In this scheduling algorithm the ready queue is organized according to the burst time of process. The routines which requires small amount of time to its completions are placed in front of queue. We have discussed three case studies. With explanation in tables and Gantt Charts. Also with turn around time and average waiting time.

For future works, I propose a more optimum algorithm of SJF, with some modification in terms the shortest time to be chosen. REFERENCES [1]. Abbas, A. A. (2015).

Multiprocessor and Real-Time Scheduling Shortest-Job-First (SJF) Scheduling Algorithm. *European Journal of Computer Science and Information Technology*, Vol 3, No. 5, 8-20.

[2]. Hasan, T. F. (2014). CPU Scheduling Visualization. *Diyala Journal of Engineering Sciences* Vol 07. No. 01 March, 16-29. [3]. Ji Young Heo, M. P. (2017).

Shortest Job First Scheduling for Reducing Jitter in Cyber Physical Systems. *International Conference Embedded Systems, Cyber-physical Systems, & applications*. [4]. Kuldeep Singh Kaswan, A. (2017).

A New Technique for CPU Scheduling: Standard Deviation Based. *International Journal of Advanced Research in Computer Engineering & Technology*, Volume 6, Issue 8, August. [5]. Mario Jean Rene, D. K. (2014).

Equitable Shortest Job First: A Preemptive Scheduling Algorithm for Soft Real-Time Systems. *International Journal of Engineering Research and Innovation*, V6, N1, Spring/Summer. [6]. Monica Santika, S. H. (2014).

Implementasi Algoritma Shortest Job First dan Round Robin pada Sistem Penjadwalan Pengiriman Barang. *Ultimatics*, Vo. VI, No. 2, December. [7]. Muhammad Akhtar, B.

H. (2015). An Optimized Shortest Job First Scheduling Algorithm for CPU Scheduling. *Journal of Applied Environmental and Biological Sciences*. [8]. Nevila Xoxa, M. Z. (2014).

Simulation of First Come First Served (FCFS) and Shortest Job First (SJF) Algorithms. *International Journal of Computer Science and Network*, Volume 3, Issue 6, December. [9]. Shweta Jain, S. J. (2016). A Review Study on CPU Scheduling Algorithms.

International Journal of Advanced Research in Computer and Communication Engineering, Vol 5. Issue 8, August.

INTERNET SOURCES:

-----

1% - [http://faratarjome.ir/u/media/shopping\\_files/store-EN-1484204753-3159.pdf](http://faratarjome.ir/u/media/shopping_files/store-EN-1484204753-3159.pdf)

1% - <https://ijarce.com/issues/volume-9-issue-4-april-2020/>

1% - <http://www.issr-journals.org/links/papers.php?journal=ijsr&application=pdf&article=IJ>

SR-14-272-17

1% -

<https://learningoperatingsystem.blogspot.com/2015/09/explain-following-scheduling-algorithms.html>

<1% -

<https://prepinsta.com/operating-systems/shortest-job-first-scheduling-preemptive-example/>

<1% - <https://www.cs.rutgers.edu/~pxk/416/notes/07-scheduling.html>

<1% - <https://www.coursehero.com/file/73644531/OS-Introduction-Extendedpptx/>

<1% -

[https://www.researchgate.net/publication/4305451\\_A\\_CPU\\_scheduling\\_algorithm\\_simulator](https://www.researchgate.net/publication/4305451_A_CPU_scheduling_algorithm_simulator)

1% - [https://www.ucstgi.edu.mm/wp-content/uploads/2020/10/CCAR\\_2018\\_20.pdf](https://www.ucstgi.edu.mm/wp-content/uploads/2020/10/CCAR_2018_20.pdf)

<1% -

[https://www.ijemr.net/DOC/AResearchSurveyAndAnalysisForCPUSchedulingAlgorithmsUsingProbabilityBasedStudy\(628-633\).pdf](https://www.ijemr.net/DOC/AResearchSurveyAndAnalysisForCPUSchedulingAlgorithmsUsingProbabilityBasedStudy(628-633).pdf)

1% -

[https://www.researchgate.net/publication/42368534\\_Performance\\_Assessment\\_of\\_Some\\_CPU\\_Scheduling\\_Algorithms](https://www.researchgate.net/publication/42368534_Performance_Assessment_of_Some_CPU_Scheduling_Algorithms)

1% -

<http://www.wseas.us/e-library/conferences/2012/Singapore/ACCIDS/ACCIDS-27.pdf>

<1% - [https://en.wikipedia.org/wiki/Scheduling\\_\(computing\)](https://en.wikipedia.org/wiki/Scheduling_(computing))

<1% -

<https://ecomputernotes.com/fundamental/disk-operating-system/batch-processing-operating-system>

1% - <https://www.everything2.com/title/CPU+scheduling>

1% -

[https://www.researchgate.net/publication/334454781\\_CPU\\_scheduling\\_visualization](https://www.researchgate.net/publication/334454781_CPU_scheduling_visualization)

<1% - <https://www.youtube.com/watch?v=-KgioB-8bqU>

<1% - <https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/>

<1% -

<https://medium.com/@bundetcom/the-concept-of-cpu-scheduler-63284a4963b6>

<1% - <https://www.grin.com/document/267035>

3% -

<http://www.eajournals.org/wp-content/uploads/Multiprocessor-and-Real-Time-Scheduling-Shortest-Job-First-SJF-Scheduling-Algorithm.pdf>

1% -

[https://www.researchgate.net/profile/Hamid\\_Hadwan/publication/330076104\\_IJARCCCE/links/5c2bea2d458515a4c7065d12/IJARCCCE.pdf](https://www.researchgate.net/profile/Hamid_Hadwan/publication/330076104_IJARCCCE/links/5c2bea2d458515a4c7065d12/IJARCCCE.pdf)

<1% - <https://www.javatpoint.com/os-process-queues>



<1% - <https://www.go4expert.com/articles/types-of-scheduling-t22307/>  
<1% -  
<https://www.slideshare.net/jyotirmaypatel/operating-system-lab-manual-44401256>  
1% - <http://plaza.ufl.edu/vejita/public.html/ExamKey.doc>  
1% -  
<https://www.coursehero.com/file/p4cg82t/Shortest-Remaining-Time-SRT-Scheduling-The-SRT-is-the-preemptive-counterpart-of/>  
1% - <http://article.sapub.org/10.5923.j.ajis.20170703.06.html>  
1% - <https://os-concepts.blogspot.com/2008/08/multilevel-queue-scheduling.html>  
1% -  
<https://www.uniassignment.com/essay-samples/information-technology/the-operating-systems-and-computer-architecture-information-technology-essay.php>  
1% - <https://research.ijcaonline.org/comnet/number1/comnet1024.pdf>  
<1% -  
<https://afteracademy.com/blog/process-scheduling-algorithms-in-the-operating-system>  
1% -  
[http://ijcsn.org/IJCSN-2014/3-6/Simulation-of-First-Come-First-Served-\\_FCFS\\_ and-Shortest-Job-First-\\_SJF\\_ Algorithms.pdf](http://ijcsn.org/IJCSN-2014/3-6/Simulation-of-First-Come-First-Served-_FCFS_ and-Shortest-Job-First-_SJF_ Algorithms.pdf)  
1% - <https://phdessay.com/simulation-of-first-come-first-served-fcfs/>  
<1% -  
[https://www.researchgate.net/publication/50194216\\_An\\_Optimized\\_Round\\_Robin\\_Scheduling\\_Algorithm\\_for\\_CPU\\_Scheduling/fulltext/0ffb719b0cf2c250b171b4e4/An-Optimized-Round-Robin-Scheduling-Algorithm-for-CPU-Scheduling.pdf](https://www.researchgate.net/publication/50194216_An_Optimized_Round_Robin_Scheduling_Algorithm_for_CPU_Scheduling/fulltext/0ffb719b0cf2c250b171b4e4/An-Optimized-Round-Robin-Scheduling-Algorithm-for-CPU-Scheduling.pdf)  
<1% -  
[https://www.oreilly.com/library/view/operating-system-concepts/9781118804926/13\\_chapter06.html](https://www.oreilly.com/library/view/operating-system-concepts/9781118804926/13_chapter06.html)  
<1% -  
<https://cs.nyu.edu/~gottlieb/courses/2000s/2002-03-fall/os/lectures/lecture-06.html>  
<1% -  
<https://www.codingalpha.com/preemptive-shortest-job-first-algorithm-c-program/>  
2% -  
[https://www.researchgate.net/publication/286452857\\_An\\_Optimized\\_Shortest\\_job\\_first\\_Scheduling\\_Algorithm\\_for\\_CPU\\_Scheduling](https://www.researchgate.net/publication/286452857_An_Optimized_Shortest_job_first_Scheduling_Algorithm_for_CPU_Scheduling)  
1% -  
[http://textroad.com/pdf/JAEBS/J.%20Appl.%20Environ.%20Biol.%20Sci.,%205\(12\)42-46,%202015.pdf](http://textroad.com/pdf/JAEBS/J.%20Appl.%20Environ.%20Biol.%20Sci.,%205(12)42-46,%202015.pdf)  
1% -  
[https://www.researchgate.net/profile/Omkar\\_Bhat2/publication/330114701\\_IoT\\_based\\_Counteracting\\_Services\\_for\\_CARDIAC\\_Patients/links/5c2e3565458515a4c70a55ea/IoT-based-Counteracting-Services-for-CARDIAC-Patients.pdf](https://www.researchgate.net/profile/Omkar_Bhat2/publication/330114701_IoT_based_Counteracting_Services_for_CARDIAC_Patients/links/5c2e3565458515a4c70a55ea/IoT-based-Counteracting-Services-for-CARDIAC-Patients.pdf)

<1% -

<https://www.slideshare.net/santhanyogesh/operating-systems-third-unit-fourth-semester-engineering>

<1% - <http://imada.sdu.dk/~jamik/dm510-19/>

<1% -

[https://www.researchgate.net/profile/Quan\\_Ha\\_Le/publication/289991693\\_Comparative\\_Performance\\_Analysis\\_of\\_PostgreSQL\\_High\\_Availability\\_Database\\_Clusters\\_through\\_Containment/links/5723cf1b08aee491cb377c2e.pdf?origin=publication\\_list](https://www.researchgate.net/profile/Quan_Ha_Le/publication/289991693_Comparative_Performance_Analysis_of_PostgreSQL_High_Availability_Database_Clusters_through_Containment/links/5723cf1b08aee491cb377c2e.pdf?origin=publication_list)

<1% - <https://girdhargopalbansal.blogspot.com/2013/05/>

<1% -

<http://www2.cs.uregina.ca/~hamilton/courses/330/notes/scheduling/scheduling.html>

1% -

<http://kresttechnology.com/krest-academic-projects/krest-major-projects/ECE/BTech%200%20Major%20ECE%20EMBEDDED%202016-17/Btech%20ECE%20Embedded%20Major%20BP%202016-17/3.%20Automated%20Irrigation%20System%20In%20Agriculture.pdf>

<1% - <http://staff.cs.psu.ac.th/iew/cs323-650/chapter6.pdf>

<1% - <https://www.edureka.co/blog/sjf-scheduling-in-c/>

<1% -

<https://ecomputernotes.com/fundamental/disk-operating-system/round-robin-scheduling>

<1% - [http://emb.incheon.ac.kr/xe/index.php?mid=emlab\\_thesis](http://emb.incheon.ac.kr/xe/index.php?mid=emlab_thesis)

1% - [http://ijarcet.org/?page\\_id=5238](http://ijarcet.org/?page_id=5238)

<1% - <http://ijarcet.org/>

1% - <https://core.ac.uk/display/60578890>

<1% - <http://sites.google.com/site/ijcsis/vol-8-no-7-oct-2010>

<1% - <http://sjifactor.com/passport.php?id=17829>